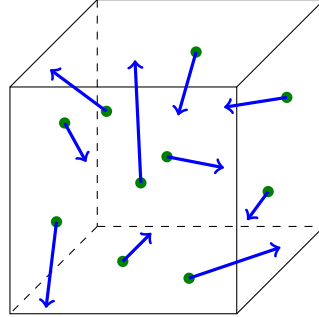


Projet Numérique Gaz Parfait

LU1MEPY3

2021-2022



1 Partie I

Un gaz parfait est une ensemble de N particules ponctuelles sans interactions formant un gaz auquel on peut associer (à l'équilibre) des variables thermodynamiques (variables d'état) comme une pression (P), un volume (V), une température (T). On considèrera que chaque particule du gaz est de nature identique aux autres et a une masse **molaire** de $m = 40 \text{ g/mol}$ pour un gaz d'Argon, cependant on suppose que le gaz n'est pas soumis à la gravité et est au repos (sans vitesse globale).

Bien que les particules qui constituent le gaz sont de même nature, cela ne signifie pas qu'elles ont toutes la même vitesse ou occupent la même position dans l'espace. Chaque particule du gaz possède donc une position $(x, y, z) = \vec{X}$ ainsi qu'une vitesse $(v_x, v_y, v_z) = \vec{v}$ qui lui sont propres.

Pour information, à l'équilibre thermodynamique avec l'enceinte qui contient le gaz parfait, considérée comme un thermostat (objet à température T constante), les vitesses des particules sont réparties selon la distribution dite de Maxwell-Boltzmann :

$$G(\vec{v}) = \left(\frac{m}{2\pi k_B T} \right)^{3/2} e^{-\frac{m|\vec{v}|^2}{2k_B T}} \quad (1)$$

Où $k_B = 1,380649 * 10^{-23} \text{ JK}^{-1}$ est la constante (exacte) de Boltzmann selon la Conférence Générale des Poids et Mesures depuis le 20 mai 2019.

Prise en main des données

Le fichier MB_N_T.txt contient les vitesses v_x, v_y, v_z d'un gaz parfait de $N \in \{10, 100, 10^3, 10^4, 10^5\}$ particules à température T en Kelvin.

1) Vérifier que $v_{th} = \sqrt{\frac{3k_B T}{m}}$ a bien la dimension d'une vitesse.

2) Importer les bibliothèques suivantes : `numpy`, `matplotlib.pyplot` (vous pouvez leur donner des alias pour faciliter leur appel plus tard dans votre code). On rappelle la syntaxe :
`import library as alias`

Si vous voulez utiliser les fonctions mathématiques de base sans appeler NumPy vous pouvez écrire : `from math import *`

[]:

3) Créer un tableau `numpy` à partir du fichier de données `MB_100_273.txt` qui contient les vitesses des particules du gaz (tableau de $N = 100$ éléments chacun étant un tableau de 3 éléments). On pourra utiliser la fonction de NumPy `loadtxt` avec l'option `delimiter=''` pour les fichiers dont les données sont séparées par des virgules (*comma-seperated values* ou *csv*)

[]:

4) Ecrire une fonction qui prend en entrée N , T et qui renvoie un tableau avec les valeurs de `MB_N_T.txt`. On rappelle qu'un nom de fichier peut contenir des variables numériques grâce à la fonction `str()` et la concaténation des chaînes de caractère. Par exemple, si $x=42$ la chaîne de caractère `'Blabla_'+str(x)` sera `'Blabla_42'`.

[]:

5) Définir v_{th} avec $T = 273$ et renormaliser le tableau par v_{th} pour travailler avec des quantités adimensionnées (créer une copie et diviser par v_{th}). Commenter l'ordre de grandeur des vitesses renormalisées.

[]:

6) Changer les dimensions de ce tableau pour qu'il contienne 3 éléments de N éléments chacun. On pourra créer un tableau `numpy` rempli de 0 avec la dimension adéquate puis le remplir des valeurs correctes repérées par deux indices dans le tableau initial v : un indice i de la ligne qui donne le numéro de la particule considéré (entre 0 et $N - 1$) et j de la colonne qui donne l'axe considéré (entre 0 et 2). On pourra aussi utiliser la fonction de NumPy `transpose`. Vérifier que les deux méthodes donnent le même résultat.

[]:

7) "Aplatir" ce tableau pour qu'il devienne un tableau à une dimension de $3N$ éléments avec la fonction `ndarray.flatten` de NumPy.

[]:

8) Afficher un histogramme de la répartition des vitesses v_x en choisissant correctement la plage de valeurs affichées.

[]:

- 9) Calculer la moyenne des vitesses (adimensionnées) pour chaque fichier avec $T = 273$. Par isotropie, on pourra par exemple faire la moyenne du tableau "aplati" et corriger d'un facteur 3 pour prendre en compte le fait qu'on doit moyenner sur N particules et non $3N$. Commenter l'influence de N sur cette grandeur.

[]:

- 10) Tracer la moyenne en fonction de $\frac{1}{\sqrt{N}}$. Ajouter des barres d'erreurs sur la vitesse moyenne dont la valeur sera bien choisie (indice: $\delta v \simeq \frac{v_{th}}{\sqrt{N}}$). Utiliser la fonction `polyfit` de NumPy pour proposer un modèle et commenter. Pour afficher les barres d'erreurs sur un graphe, il suffit d'utiliser la fonction de `matplotlib` nommée `errorbar` qui prend en entrée les tableaux contenant respectivement les abscisses, les ordonnées et les erreurs selon les axes x ou y . Par exemple on pourra appeler : `plt.errorbar(abscisses, ordonnees, yerr = err_ordonnees)`

[]:

2 Partie II

On a étudié les propriétés du gaz parfait à partir d'une "photographie" de la distribution des vitesses à un instant t . On souhaite à présent simuler une dynamique du système de manière simplifiée en supposant que toutes les vitesses sont des vecteurs de norme v_{th} orientés selon $\pm \vec{u}_z, \vec{u}_x, \vec{u}_y$. Dans ce modèle les vitesses des N particules sont réparties de manière uniforme selon ces 6 directions qui sont exclusives les unes des autres (le vecteur vitesse d'une particule est dirigée selon $+\vec{u}_z$ ou $-\vec{u}_z$ ou $+\vec{u}_y$...).

- 1) Ecrire une fonction qui crée un tableau contenant les positions des N particules constituant le gaz. Ces positions (3 coordonnées) seront choisies aléatoirement au sein d'une boîte cubique de côté 1m. On pourra utiliser notamment la fonction `random()` de la librairie du même nom qui renvoie un nombre aléatoire de $[0; 1[$

[1]: `import random as rd`

- 2) Adapter votre fonction précédente pour qu'elle écrive un fichier contenant les positions des particules. Les différentes valeurs selon les différents axes seront séparées par une virgule pour respecter le format *csv*. De plus le fichier devra contenir une ligne par particule, il faudra donc sauter de ligne une fois les trois coordonnées d'une particule écrites. Pour ce faire on utilisera la chaîne de caractère `"\n"` qui correspond à un saut de ligne.

[]:

- 3) Créer une fonction qui écrit un fichier de vitesses correspondant au modèle hexadirectionnel et renvoie le tableau correspondant. On pourra tirer profit de la fonction `choice` du module `random` qui permet de choisir aléatoirement une des valeurs de la séquence donnée en argument. Par exemple, `choice((-1, 1))` renvoie aléatoirement -1 ou 1 . Il faudra ainsi tirer aléatoirement la direction de la vitesse (indice entre 0 et 2) ainsi que le sens de la vitesse (± 1).

[]:

Prévoir au moins une séance pour la partie III, si vous n'avez pas fini les dernières questions de cette partie avant la fin de la deuxième séance passez directement à la suite quitte à y revenir plus tard.

- 4) Afficher une représentation des positions de quelques particules parmi les N (afficher les toutes si $N < 30$). Pour les N plus grands (supérieurs à 30), vous sélectionnerez les particules d'indice multiple de 3 inférieur à 100 et en extrairez les coordonnées dans un nouveau tableau dont vous déterminerez la taille. Vous ferez bonne usage des conditions if pour distinguer les cas des grands N et petits N ainsi que pour sélectionner les indices.

L'affichage 3D se fait avec la fonction Axes3D de la librairie `mpl_toolkits.mplot3d`. Les principales lignes de codes sont pré-écrites, il ne reste qu'à remplir le nuage de points (`scatter`) avec les tableaux des coordonnées selon les différents axes (attention l'ordre compte).

[]:

```
[2]: from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = Axes3D(fig)

ax.scatter(, , , c='r')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
plt.show()
```

- 5) Faire évoluer les positions et les vitesses à un instant $t + \delta t$ avec comme condition que si une particule touche une paroi, elle est réfléchi ce qui transforme \vec{v} en $-\vec{v}$. On choisira $\delta t = 10^{-3}$ s. Afficher les positions avant et après évolution.

[]:

Question optionnelle

Ecrire une fonction `GP(N, T)` qui génère un fichier texte contenant les vitesses des particules du gaz parfait selon la distribution de Maxwell-Blotzmann et qui renvoie le tableau correspondant. m et k_B seront des paramètres à définir au sein de la fonction. On pourra notamment utiliser la fonction `gauss` de la librairie `random` (à importer).

[]:

3 Partie III

On peut maintenant s'intéresser aux caractéristiques thermodynamiques du système. En effet, outre les quelques variables thermodynamiques qu'on peut associer à un système, à l'équilibre on peut également définir des fonctions d'état qui sont fonction des variables thermodynamiques et qui ne dépendent que de l'état actuel du système. En particulier la différence d'une fonction prise entre deux états ne dépend pas de la transformation (réelle ou hypothétique) parcourue pour passer d'un état à l'autre.

L'énergie interne U est une telle fonction d'état dite extensive, c'est à dire proportionnelle au nombre de particule constituant le gaz parfait.

Dans le cas du gaz parfait, on peut relier l'énergie interne à la distribution des vitesses et à la température par les relations :

$$U = \frac{1}{2}mN\langle v^2 \rangle = \frac{3}{2}Nk_B T \quad (2)$$

Où $\langle \rangle$ signifie moyenne sur les particules du gaz.

Ces relations rendent explicitement compte de l'extensivité de l'énergie interne et permettent de faire le lien entre l'agitation thermique microscopique des particules (la moyenne de v^2) et la température via le théorème d'équipartition de l'énergie vu plus tard dans vos études.

- 1) Pour un fichier de votre choix, calculer la moyenne de v^2 , calculer également la variance de v et commenter. On fera attention à la correction par un facteur 3 déjà évoquée plus haut.

[]:

- 2) En adaptant votre code de la question I.9), évaluer l'influence de N sur $\langle v^2 \rangle$ (à T fixée). Commenter. Qu'en est-il de l'influence de la température (à N fixé)? Vous prendrez garde à la normalisation qui peut jouer un rôle ici (faut-il ou non normaliser par v_{th} , v_{th}^2 , ...?).

[]:

- 3) Créer un tableau qui regroupe les variances des vitesses pour chaque fichier. Ce sera donc un tableau à deux indices correspondant aux valeurs possibles pour (N, T) .

[]:

- 4) Adapter le code précédent pour qu'il renvoie une liste (tableau précédent "aplati") qui correspond aux valeurs de l'énergie interne U . On prendra soin de noter les différentes valeurs de $N * T$ dans une liste (ou tableau numpy de dimension 1) lors de la construction du tableau.

[]:

- 5) Afficher U en fonction de $N * T$ et retrouver la valeur de la capacité calorifique molaire à volume constant. On rappelle que $\left. \frac{\partial U}{\partial T} \right|_{N,V} = nC_{V,m} = C_V$. La capacité calorifique molaire à volume constant correspond donc au coefficient directeur du graphe à tracer multiplié par le nombre d'Avogadro \mathcal{N}_A . Pour réaliser l'ajustement on pourra utiliser la fonction `polyfit`.

[]: